Freeform Search

Database: US Pre-Grant Publication Full-Text Database; US Patents Full-Text Database; US OCR Full-Text Database; EPO Abstracts Database; JPO Abstracts Database; Derwent World Patents Index; IBM Technical Disclosure Bulletins

Term: L13 or L11

Display: 50 **Documents in Display Format:** REV **Starting with Number** 1

Generate: ○ Hit List ◉ Hit Count ○ Side by Side ○ Image

Search | Clear | Interrupt

Search History

DATE: Thursday, December 20, 2007 **Purge Queries** Printable Copy Create Case

Set Name side by side	Query	Hit Count	Set Name result set
DB=USPT; PLUR=NO; OP=OR			
L15	L13 or L11	38	L15
L14	L8 and (test adj execution adj report)	0	L14
L13	L8 and (test adj execution)	11	L13
L12	L8 and (source adj code adj management)	0	L12
L11	L8 and (configuration adj management)	35	L11
L10	L8 and ter	0	L10
L9	L8 and scm	0	L9
L8	L7 and (coverage or path or paths)	52	L8
L7	L5 and interface	64	L7
L6	L5 and (API or (application ADJ program adj interface))	54	L6
L5	L4 AND (source adj code)	65	L5
L4	L3 or L2	127	L4
L3	bug near(identification or identifier)	11	L3
L2	bug same (identification or identifier)	127	L2
L1	bug and (identification or identifier)	2002	L1

Static Source Code Analysis Tools for C

- **The Leading Commercial Tools**
  - CodeSonar (Grammatech)
    A new member of the CodeSurfer family (see below). The tool is very effective in spotting code defects, and suspicious code fragments. The tool is currently being extended with a rule checker for set of coding rules for safety critical code, which will make it extra attractive for high integrity applications. Especially good at intra-procedural analysis. Highly recommended.
  - Coverity
    Leading edge tool based on Dawson Engler's methodology for source code analysis of large code bases. An extended version of the tool supports user-defined properties in the Metal language. Very few false positives, but can be expensive.
  - KlocWork
    Support for static error detection, with added project management and project visualization capabilities. Fast, almost as thorough as Coverity, and not quite as expensive. Especially good at finding array bound violations. A capability for user-defined checks is pending.
  - PolySpace
    Marketed by a French company co-founded by students of Patrick Cousot (pioneer in the area of abstract interpretation). Polyspace claims it can intercept 100% of the runtime errors in C programs. (See cverifier.htm.) Customers are in the airline industry and the European space program. Can be thorough, but also very slow, and does not scale beyond a few thousand lines of code. Does not support full ANSI-C language (e.g., it places restrictions on the use of gotos).
  - Purify (Rational)
    This tool is focused primarily on the detection

of memory leaks, and not on general source code analysis. It is used fairly broadly.
- The Lint family, e.g. PC-Lint/FlexeLint (Gimpel), Lint Plus (Cleanscape)
  Generic source code analysis, value tracking, some types of array indexing errors. Suffers from high, sometimes very high, false positive rates, but the output can be customized with flags and code annotations.
- PREfix and PREfast (Microsoft)
  Effective, but Microsoft proprietary, tools. PREfix was developed by Jon Pincus; MicroSoft acquired the tool when it bought Pincus' company. PREfast is a lighter weight tool, developed within Microsoft as a faster alternative to PREfix (though it is not based on PREfix itself). Both these tools are reported to be very effective in intercepting defects early, and come with filtering methods for the output to reduce the false positive ratio. PREfast allows for new defect patterns to be defined via plugins. Less than 10% of the code of PREfix is said to concern with analysis per se, most applies to the filtering and presentation of output, to reduce the number of false positives.
- Safer C (Oakwood Computing)
  Based on L. Halton's 1995 book on Safer C, now out of print, covering code analysis and enforcement of coding guidelines. Not found to be too useful in our tests.

- **Academic and Research tools**

  - Calysto (new) a tool by Domagoj Babic
  - Saturn (new) by Alex Aiken and others at Stanford.
  - mygcc (new)
    An extension of the gcc compiler supporting user-defined checks written in a simple formalism, that can be checked efficiently. Path queries can be run on the control-flow graph of functions, specifying a start node, a stop node, and constraints on the path in between. Sample queries are provided that match many of the ones used in Engler's early

study using the Metal language (now supported by the Coverity tool). A very interesting project that will hopefully find its way into the main gcc distribution soon.

- ESC (Compaq/HP)
  Extended static checker for Java and for Modula3. developed by Greg Nelson and colleagues, which is based on a mix of theorem proving and static analysis. It's thorough and effective, but also slow, and needs considerable knowledge to run. This tool does not target C, and therefore does not properly belong in this listing, but we include it as one of the landmark research tools in this domain.
- LC-Lint
  The descendent of the early research Unix version of lint, which was written by Steve Johnson in 1979. This tool needs lots of annotations to work well, and often produces overwhelming amounts of output.
- Vault (MicroSoft)
  An experimental system, in development at MicroSoft by Rob DeLine and Manuel Fahndrich. It is based on formal annotations placed in the code.
- Astree (CNRS, France)
  Astree is a static program analyzer for structured C programs, but without support for dynamic memory allocation and recursion (as used, for instance for embedded systems and in safety critical systems). The tool name is an acronym for Analyseur statique de logiciels temps-reel embarques (static analyzer for real-time embedded software). Among those working on this tool are Patrick and Radhia Cousot.
- CGS (C Global Surveyor, NASA ARC)
  An experimental tool at NASA Ames Research Center from Guillaume Brat and Arnaud Venet, based on abstract interpretation techniques, inspired by Patrick Cousot. Work in progress.
- C-Kit (Bell Labs).
  A research toolkit developed at Bell Labs, with algorithms for pointer alias analysis, program slicing, etc. for ANSI C. Written in SML. Can produce parsetree and symbol table

information, but, as yet, not call flow graphs or function call graphs. The principal researchers involved in this work (Nevin Heintze, Jon Riecke, Dave MacQueen) are no longer at Bell Labs and development has stopped.
  - Uno (Bell Labs)
    Lightweight tool for static analysis. The tool is targeted at a small set of common programming defects (Uninitialized data, Nil-pointer dereferencing, and Out-of-bound array indexing, with the three initial letters giving the tool its name). It also handles a range of simple, user-defined properties.
  - Orion (Bell Labs)
    Work in progress on an extension of Uno for C++, based on gcc.

- **Other tools (Code Browsers; Development Environments)**
  - Programming Style or Guidelines Checkers
    - Parasoft CodeWizard
    - Plum Hall SQS
    - QA C
    - CheckMate
  - CodeSurfer
    Supports data dependence analysis, program slicing for C, interprocedural flow analysis. The company was co-founded by Tom Reps. Very well done GUI. Mostly research applications. See also CodeSonar above.
  - Semantic Designs
    Offers front-ends for many different languages, Supports some flow analysis. Geared towards code transformations or re-engineering. Targets large code bases.

- **Links**
  - http://testingfaqs.org/t-static.html
  - Short white paper on the construction of static analyzers
  - Another overview of static analyzers
  - Up-to-date overview of source code checkers
  - Wikipedia entry (new)

Recommended Textbooks

F. Nielson, H. R. Nielson and C. Hankin,
Principles of Program Analysis,
ISBN 3-540-65410-0, Springer-Verlag.

last update 17 November 2007 back to spinroot.com

IBM

Redbooks Search

Most IBM Redbooks are available in Adobe PDF format and may be viewed online or downloaded for offline viewing and printing. Read our How to buy page for details on how to order a bound hardcopy . The following search result is for Redbooks only. To search the entire web site, use the search box in the upper right corner.

Redbooks - Search Results

test coverage Search

Use AND, OR, NOT to separate keywords

0 results found

Back to Index

IBM

Redpapers™ Index

Redpapers are technical documents that have been written to address a specific topic. They are not planned to become published redbooks and are not necessarily the result of an ITSO residency.

Note that these documents are not orderable in hardcopy. Since these papers are not published through our normal publishing process, the layouts may vary.

Redpapers - Search Results

purify Search

Use AND, OR, NOT to separate keywords

0 results found

Back to Index